Supplementary Material for Superpixel Convolutional Networks using Bilateral Inceptions

Raghudeep Gadde[1⋆], Varun Jampani[2⋆], Martin Kiefel[2,3], Daniel Kappler[2], and Peter V. Gehler[2,3]

[1]Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE, UPEM, France
[2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[3]Bernstein Center for Computational Neuroscience, Tübingen, Germany

In this supplementary, we first discuss the use of an approximate bilateral filtering in BI modules (Sec. 1). Later, we present some qualitative results using different models for the approach presented in the main paper (Sec. 2).

1 Approximate Bilateral Filtering

The bilateral inception module presented in the main paper computes a matrix-vector product between a Gaussian filter K and a vector of activations $\mathbf{z}_c$. Bilateral filtering is an important operation and many algorithmic techniques have been proposed to speed-up this operation [1,2,3]. In the main paper we opted to implement what can be considered the brute-force variant of explicitly constructing K and then using BLAS to compute the matrix-vector product. This resulted in a few millisecond operation. The explicit way to compute is possible due to the reduction to super-pixels, e.g., it would not work for DenseCRF variants that operate on the full image resolution.

Here, we present experiments where we use the fast approximate bilateral filtering algorithm of [2]. This choice allows for larger dimensions of matrix-vector multiplication. The reason for presenting the explicit multiplication in the main paper was that it was computationally faster. For the small sizes of the involved matrices and vectors, the explicit computation is sufficient and we had no GPU implementation of an approximate technique that matched this runtime. Also it is conceptually easier and the gradient to the feature transformations ($\Lambda\mathbf{f}$) is obtained using standard matrix calculus.

1.1 Experiments

We modified the existing segmentation architectures analogous to the main paper. The main difference is that, here, the inception modules use the lattice approximation [2] to compute the bilateral filtering. Using the lattice approximation did not allow us to back-propagate through feature transformations (Λ)

[1] The first two authors contribute equally to this work.

and thus we used hand-specified feature scales as will be explained later. Specifically, we take CNN architectures from the works of [4,5,6] and insert the BI modules between the spatial FC layers. We use superpixels from [7] for all the experiments with the lattice approximation. Experiments are performed using Caffe neural network framework [8].

Model	IoU	Runtime
DeepLab	68.9	145ms
$BI_7(2)$-$BI_8(10)$	**73.8**	+600
DeepLab-CRF [4]	72.7	+830
DeepLab-MSc-CRF [4]	**73.6**	+880
DeepLab-EdgeNet [9]	71.7	+30
DeepLab-EdgeNet-CRF [9]	**73.6**	+860

Table 1. Semantic Segmentation using DeepLab model. IoU scores on the Pascal VOC12 segmentation test dataset with different models and our modified inception model. Also shown are the corresponding runtimes in milliseconds. Runtimes also include superpixel computations (300 ms with Dollar superpixels [7])

Model	IoU
CNN	67.5 / -
DeconvNet (CNN+Deconvolutions)	69.8 / 72.0
$BI_3(6)$-$BI_4(6)$-$BI_7(2)$-$BI_8(6)$	71.9 / -
$BI_3(6)$-$BI_4(6)$-$BI_7(2)$-$BI_8(6)$-G(6)	73.6 / **75.2**
DeconvNet-CRF (CRF-RNN) [5]	73.0 / 74.7
Context-CRF-RNN [10]	- / **75.3**

Table 2. Semantic Segmentation using CRFasRNN model. IoU score corresponding to different models on Pascal VOC12 reduced validation / test segmentation dataset. The reduced validation set consists of 346 images as used in [5] where we adapted the model from.

Semantic Segmentation The experiments in this section use Pascal VOC12 segmentation dataset [11] with 21 object classes and the images have a maximum resolution of 0.25 megapixels. For all experiments on VOC2012, we train using the extended training set of 10581 images collected by [12]. We modified the DeepLab network architecture of [4] and CRFasRNN architecture from [5] which uses CNN with deconvolution layers followed by DenseCRF trained end-to-end.

DeepLab Model We experimented with $BI_7(2)$-$BI_8(10)$ inception model. Results using the DeepLab model are summarized in Tab. 1. Although we get similar improvements with inception modules as with the explicit kernel computation, using lattice approximation is slower.

CRFasRNN Model We add BI modules after score-pool3, score-pool4, FC_7 and FC_8 1×1 convolution layers resulting in $BI_3(6)$-$BI_4(6)$-$BI_7(2)$-$BI_8(6)$ model and also experimented with another variant where BI_8 is followed by a another inception module, G(6), with 6 Gaussian kernels. Note that here also we discarded both deconvolution and DenseCRF parts of the original model [5] and inserted the BI modules in the base CNN and found similar improvements compared to the inception modules with explicit kernel computaion. See Tab. 2 for results on the CRFasRNN model.

Model	Class / Total accuracy
AlexNet CNN	55.3 / 58.9
$BI_7(2)$-$BI_8(6)$	68.5 / 71.8
$BI_7(2)$-$BI_8(6)$-G(6)	67.6 / 73.1
AlexNet-CRF	65.5 / 71.0

Table 3. Material Segmentation using AlexNet. Pixel accuracy of different models on MINC material segmentation test dataset [6].

Material Segmentation Table 3 shows the results on MINC dataset [6] with modifying the AlexNet architecture with our inception modules. We observe similar improvements as with explicit kernel construction. For this model, we do not provide any learned setup due to very limited segment training dataset. The weights to combine outputs in the bilateral inception layer are found by validation on the validation set.

Scales of Bilateral Inception Modules Unlike the technique presented in the main paper, we didn't back-propagate through feature transformation (Λ) using the approximate bilateral filter technique. So, the feature scales are hand-specified and validated, which are as follows. The optimal scale values for the $BI_7(2)$-$BI_8(2)$ model are found by validation for the best performance which are σ_{xy} = (0.1, 0.1) for the spatial (XY) kernel and σ_{rgbxy} = (0.1, 0.1, 0.1, 0.01, 0.01) for color and position (RGBXY) kernel. Next, as more kernels are added to $BI_8(2)$, we set scales to be $\alpha^*(\sigma_{xy}, \sigma_{rgbxy})$. The value of α is chosen as 1, 0.5, 0.1, 0.05, 0.1, at uniform interval, for the $BI_8(10)$ bilateral inception module.

2 Qualitative Results

In this section, we present more qualitative results obtained using BI module with explicit kernel computation technique presented in the main paper. Results on Pascal VOC12 dataset [11] using the DeepLab-LargeFOV model are shown in Fig. 1, followed by the results on MINC dataset [6] in Fig. 2 and on Cityscapes dataset [13] in Fig. 3.

References

1. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: European conference on computer vision, Springer (2006) 568–580
2. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. In: Computer Graphics Forum. Volume 29., Wiley Online Library (2010) 753–762

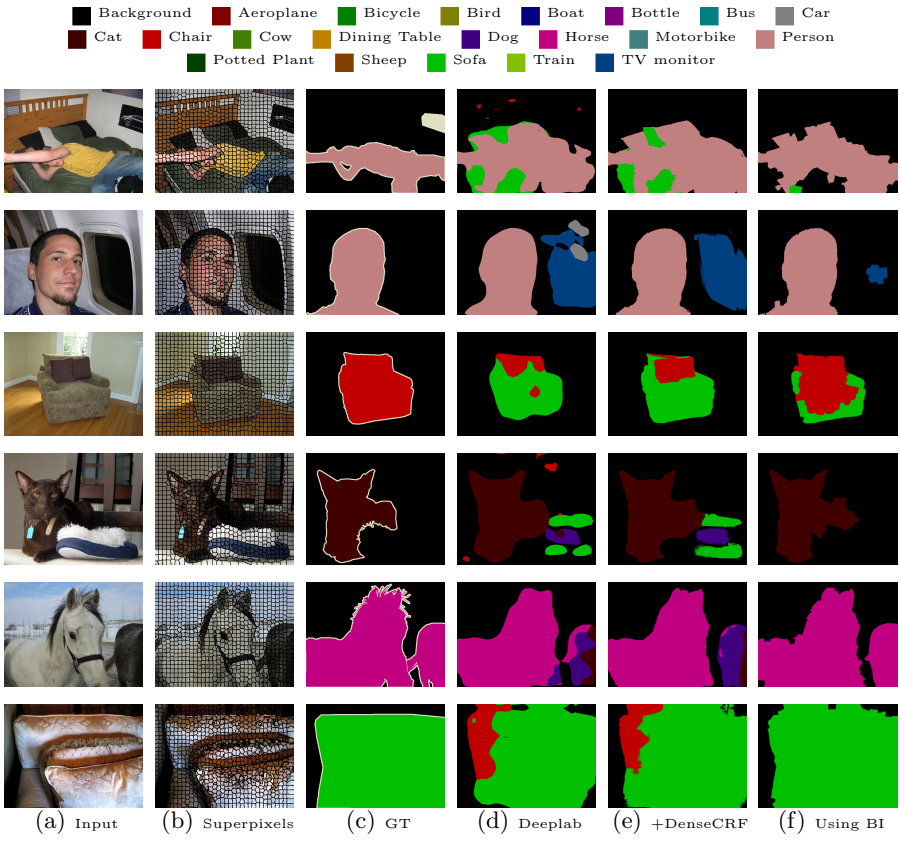

Fig. 1. Semantic Segmentation. Example results of semantic segmentation on Pascal VOC12 dataset. (d) depicts the DeepLab CNN result, (e) CNN + 10 steps of mean-field inference, (f) result obtained with bilateral inception (BI) modules ($BI_6(2)$+$BI_7(6)$) between FC layers.

3. Gastal, E.S., Oliveira, M.M.: Domain transform for edge-aware image and video processing. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 69
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: International Conference on Learning Representation. (2015)
5. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1529–1537
6. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3479–3487
7. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1841–1848

Fig. 2. Material Segmentation. Example results of material segmentation. (d) depicts the AlexNet CNN result, (e) CNN + 10 steps of mean-field inference, (f) result obtained with bilateral inception (BI) modules ($\text{BI}_7(2)+\text{BI}_8(6)$) between FC layers.

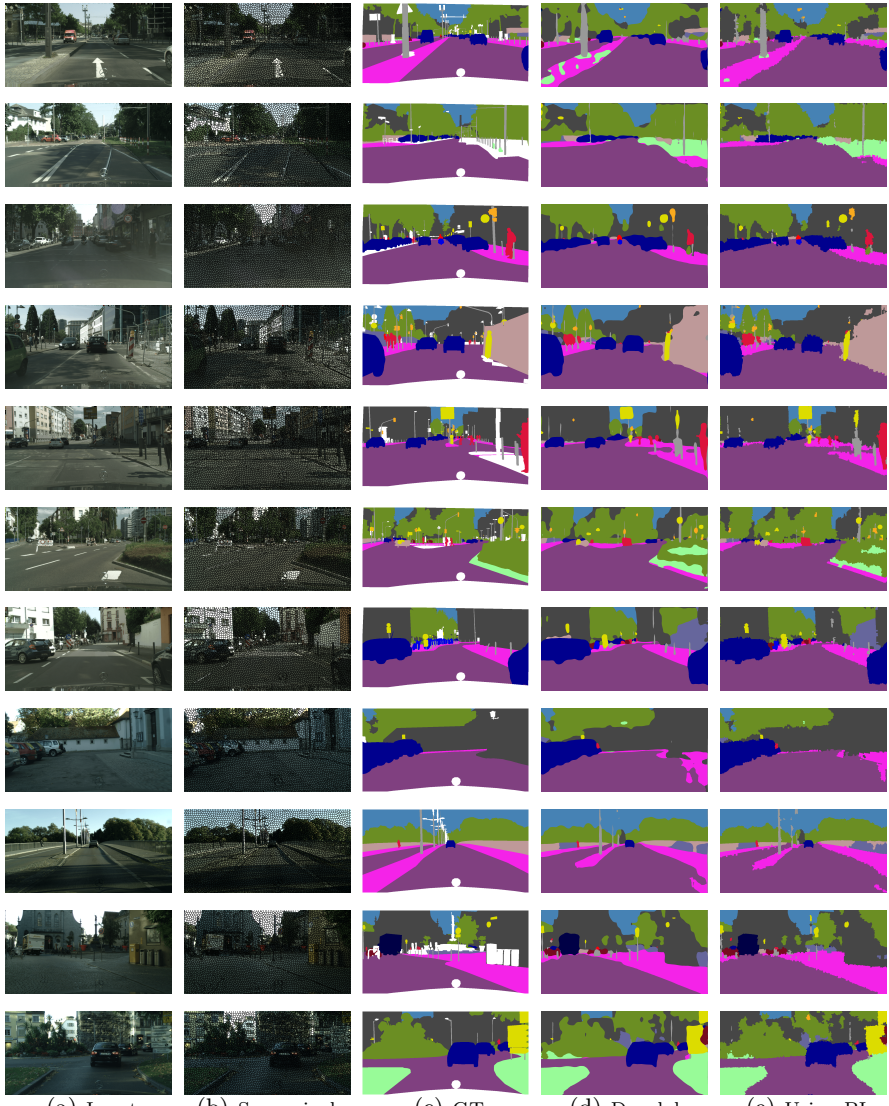

Fig. 3. Street Scene Segmentation. Example results of street scene segmentation. (d) depicts the DeepLab results, (e) result obtained by adding bilateral inception (BI) modules ($BI_6(2)$+$BI_7(6)$) between FC layers.

8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM (2014) 675–678

9. Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4545–4554
10. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representation. (2016)
11. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL VOC2012 challenge results. (2012)
12. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: Proceedings of the IEEE International Conference on Computer Vision. (2011) 991–998
13. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset. In: CVPR Workshop on The Future of Datasets in Vision. (2015)